Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 29 (2012) 329 – 333

Procedia Engineering

www.elsevier.com/locate/procedia

2012 International Workshop on Information and Electronics Engineering (IWIEE)

Design of USB-CAN Controller Based on PIC18F4580

Junhao Niu[a,b,]*

[a]*School of Electronic Engineering and Automation, Guilin University of Electronic Technology, Guilin, Guangxi, 541004, China*
[b]*School of Electronical & Machanical Engineering，Xidian University，Xi'an 710071，China*

Abstract

A designed method of USB-CAN converter with PIC18F4580 is proposed in this paper. A CH372 is used to communicate computer and microcontroller as a bridge. The CAN interface controller is realized by the ECAN module embedded in PIC18F4580. PCA82C50, one chip of CAN bus driver, is used as interface between the CAN protocol controller and the physical bus. A dynamic link library is designed with Visual Studio 2003 platform. With the converter and the DLL, a computer could work with a CAN network as one CAN node.

Keywords: USB-CAN controller; PIC18F4580; CH372; PCA82C50; DLL

1. Introduction

CAN (Controller Area Network) bus is a local area network developed firstly for the field of automotive and industrial automation control. Owing to its high performance at communication speed, real-time, reliability, and low cost, the bus is developed rapid and widely used in the automotive, medical, manufacturing site, etc. The requirement of information transfer is increasing with the demands of automation. RS232, RS485 and other master-slave bus can not meet the requirements, and the CAN bus as the station communication network will significantly improve the performance of the system, making the system under normal operation in the harsh environment.

When building an automatic control system with computer and CAN bus devices, one convert device is often needed as the bridge between computer and CAN networks. This paper presents a design method of USB-CAN converter based on PIC18F4580 microcontroller and USB interface chip CH372. Using the controller, a computer can access and control terminal equipments on the CAN network as a CAN node.

* Corresponding author. Tel.: +86-0773-2191029; fax: +86-0773-2191029.
E-mail address: junhao@guet.edu.cn.

doi:10.1016/j.proeng.2011.12.717

2. System Block Diagram

The block diagram of USB-CAN converter is shown in Fig. 1. The system mainly consists of five parts: computer, USB interface circuit, micro-controller, optical isolation circuit, CAN bus driver circuit. The computer sends data through its USB interface. After receiving the data package, USB interface circuit will active the interrupt of microcontroller. The microcontroller converts, packages the data, and send to CAN bus network. The CAN bus nodes make the appropriate action according to the command, and then send back information via the same route to the computer.

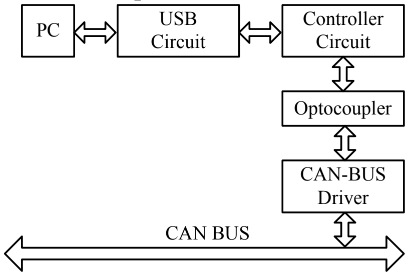

Fig. 1 System Block Diagram

3. Design of Hardware

3.1. Selecting of Controller

Microchip's enhanced 8-bit controller PIC18F4580 is used in this paper. PIC18F4580 uses 16-bit RISC instruction set, and provides the A/D converter, internal EEPROM memory, the comparator output, capture input, PWM output (with a simple filter circuit can be made after the D/A output), IIC bus and SPI bus interface circuit and many other functions. PIC18F4580 can be easily repeated programming and online debugging. PIC18F4580 has five general-purpose I/O ports that some I/O pins can be re-used for the external module [1].

PIC18F4580 devices contain an Enhanced Controller Area Network (ECAN) module. The ECAN module is a communication controller, implementing the CAN 2.0A or B protocol as defined in the BOSCH specification. The module will support CAN 1.2, CAN 2.0A, CAN 2.0B Passive and CAN 2.0B Active versions of the protocol. The CAN bus module consists of a protocol engine and message buffering and control. The CAN protocol engine automatically handles all functions for receiving and transmitting messages on the CAN bus. Messages are transmitted by first loading the appropriate data registers. Status and errors can be checked by reading the appropriate registers. Any message detected on the CAN bus is checked for errors and then matched against filters to see if it should be received and stored in one of the two receive registers [1].

3.2. Design of USB Interface Circuit

CH372 is a USB bus universal device interface chip. It is upgrade production of CH371, and function predigested edition of CH375. In the location, CH372 has 8-bit data bus and read strobe input, write strobe input and interrupt output. It is convenient to connect to CH372 on controller system bus such as CPU, DSP, MCU, MPU, and so on. In computer system, the equipped software of CH372 supplies operation interface which is handy and wieldy. When CH372 communicates with local MCU is just like read and write file [2].

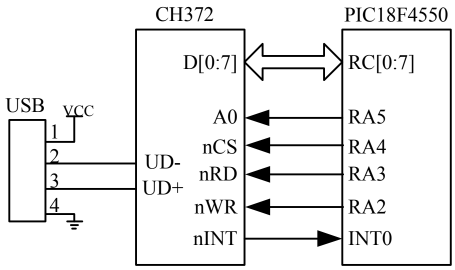

Fig. 2 USB Interface Circuit Diagram

USB bus interface circuit is shown in Fig. 2. The RC[0:7] of PIC18F4580 is connected to D[0:7], which is the data bus of CH372. RA4 pin is connected to nCS as a chip select signal. Connected with A0, RA5 is used to determine that the transfer data is command or just data. RA5 is high, the operation command port of CH372, CH372 low data port is operating. RA2 is connected with nWR in CH372 as write strobe output pin. RA3 is connected with nRD in CH372 as read strobe output pin. RB0 (INT0) connected to the interrupt output of CH372, and is active with low-level.

3.3. Design of CAN Bus Driver Circuit

The PCA82C250 is the interface between the CAN protocol controller and the physical bus. It is primarily intended for high-speed applications (up to 1 Mbaud) in cars. The device provides differential transmit capability to the bus and differential receive capability to the CAN controller. It is fully compatible with the “ISO 11898” standard [3-4].

A current limiting circuit protects the transmitter output stage against short-circuit to positive and negative battery voltage. Although the power dissipation is increased during this fault condition, this feature will prevent destruction of the transmitter output stage. If the junction temperature exceeds a value of approximately 160 ℃, the limiting current of both transmitter outputs is decreased. Because the transmitter is responsible for the major part of the power dissipation, this will result in reduced power dissipation and hence a lower chip temperature. All other parts of the IC will remain in operation. The thermal protection is particularly needed when a bus line is short-circuited. The CANH and CANL lines are also protected against electrical transients who may occur in an automotive environment [3].

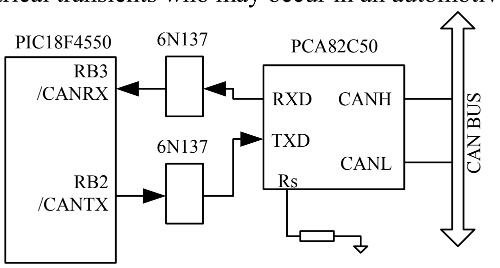

Fig. 3 CAN bus Driver Circuit

The CAN bus driver circuit is shown in Fig. 3. Two high-speed optocoupler 6N137 are used for galvanic isolation between CAN bus driver and controller, while reducing the introduction of interference from other devices, on the other hand protect the CAN high-speed communications. Pin 8 (Rs) allows three different modes of operation to be selected: high-speed, slope control or standby. The slope is proportional to the current output at pin 8. By change the value of resistor connected from pin 8 to ground, the rise and fall slope of CAN bus can be adjusted. The resistor generally between 16 ~ 140kΩ for lower speeds or shorter bus length, and unshielded twisted pair or a parallel pair of wires can be used for the bus.

4. Design of Software

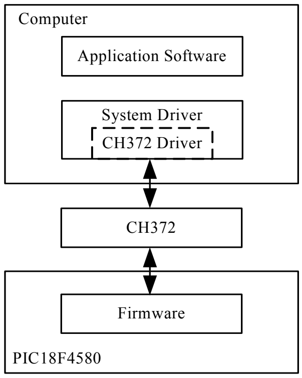

Fig. 4 Software Structure

System software architecture is shown in Fig. 4, including firmware, system driver, and application. The firmware mainly used to complete system initialization, send and receive bus data. The driver provide control interface for application. The application realizes the access or control functions according to the requirement of special system.

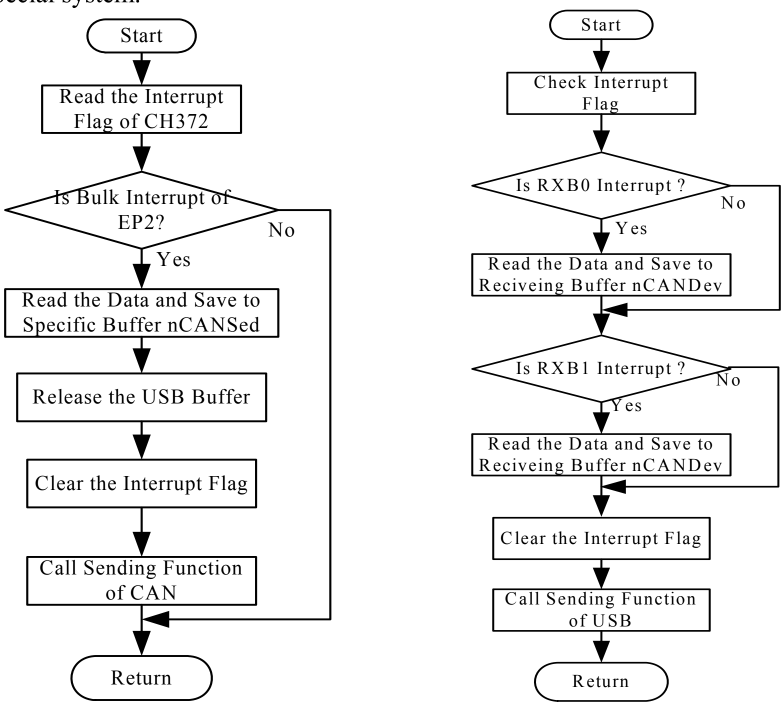

Fig. 5. (a)The Interrupt Flow of USB; (b) The Interrupt Flow of CAN

4.1. Firmware

Firmware mainly consists of initialization process, main loop and interrupts service routine.

The main loop is an infinite loop of function, which is mainly waiting the interrupt of both the USB and CAN bus.

Initialization process is manly used to configure the environment and the various registers of the hardware, allowing each module to enter the normal working condition. The process mainly includes the following steps:

CH372 initialization: Firstly, configure the I/O port mode, reset the CH372. Secondly, test CH372 to make sure that it woks well. Finally, configure the USB mode and enable INT0 interrupt.

Interrupt service function includes USB interrupt request and CAN bus module interrupt request. USB interrupt response process is shown in Fig. 5.(a), and CAN bus module response process is shown in Fig. 5.(b).

4.2. System Driver

CH372 supplies application layer interface in computer port. The application layer interface is API which is supplied by Dynamic Link Library DLL face to function of CH372 [6]. The API supplied by Dynamic Link Library of CH372 contains: device manage API, data transfer API and interrupt handle API [5].

In the paper, CH372 is not the object operated, but the CAN device is, so a dynamic link library host is re-packaged for the CAN system design. The API function provided by dynamic link library includes: open the device, close the device, configure the CAN device, reads the uploaded message, send message down to CAN network.

5. Summary

In this paper, a design method of USB-CAN converter based on PIC controller and CH372 is presented. By using the converter, computer could work in the CAN networks as a device node. The windows driver is programmed to make it easy to using this converter. Experiments show that the use of PIC18F4580 can achieve high resolution, fast speed, to meet the complex requirements of high transmission speed of the occasion.

References

[1] Microchip Technology Inc. PIC18F2480/2580/4480/4580 Data Sheet Enhanced Flash Microcontrollers with ECAN™ Technology, 10-Bit A/D and nanoWatt Technology, *Microchip Technology Inc.* 2009.

[2] Nanjing Qinheng Electronics Co., Ltd. The DataSheet of CH372, *Nanjing Qinheng Electronics Co*. July 2008.

[3] Philips Semiconductors. PCA82C250 CAN controller interface datasheet. *Philips Semiconductors*. Jan 2000.

[4] Chen Jingqian, He Zhiwei, Liu Wenze. Design of Protocol Conversion Unit between RS2232 and CAN Bus. *ELECTRONIC ENGINEER*. 2004.30(4), p. 63-66.

[5] Wang Junbo, Zou Jijun, Rao Yuntao, Wang Jinhong, Li Shuanming. The Design ofUSB Bus and CAN Bus ProtocolAdaptor. *Journal of East China Institute of Technology*. 2006. 29(3), p. 273-276.

[6] Li Lan, Ning Yonghai. The design of USB data acquisition based on CH372. *Control System*. 2007.23(12), p. 76-77.